

A NEW MODULO n MULTIPLICATION ALGORITHM WITH MODERATE FACTORS OF ($2n+2$) and ($2n+6$)

A. UMA MAHESWARI & PRABHA DURAIRAJ

Associate Professor, Quaid-E-Millath Government College for Women (Autonomous),
Chennai, Tamil Nadu, India

ABSTRACT

Security is an important technique for many applications including private networks, e-commerce and secure internet access. Public key cryptosystems like the RSA cryptosystem and the El-Gamal cryptosystem are popular security techniques. These cryptosystems have to perform modular exponentiation with large exponent and modulus for security considerations. Modular exponentiation is performed by repeated modular multiplications. This paper proposes an improvised algorithm for modulo n multiplication, where n is odd.

- The remainder with modulus n is derived from the remainders with modulus ($2n+2$) and ($2n+6$)
- ($2n+2$) and ($2n+6$) can be decomposed into products of relatively prime factors even if n is prime or difficult to be factorized into prime factors.

The efficiency of the proposed algorithm is estimated. On comparing the computational complexity with the conventional method, the new improvised algorithm is more efficient.

KEYWORDS: RSA Cryptosystem, El-Gamal Cryptosystem, Modular Multiplication, Chinese Remainder Theorem

1. INTRODUCTION

The exponentiation computation $x^c \bmod n$ plays a pivotal role in many public key cryptosystems like the RSA cryptosystem [1], El-Gamal cryptosystem [2], Rabin cryptosystem [3], Diffie-Hellman Key Exchange System [4] to list a few. To improve the speed of the exponentiation computation for large numbers with 200 to 700 digits, it is desired to devise faster encryption and decryption operations. Hayashi et al [5], Rao et al [6 and 7] and Hwang et al [8] proposed new modular multiplication algorithms to reduce the computation time of RSA. In paper [5], the remainder with modulus n is constructed from remainders with moduli ($n+1$) and ($n+2$).

In paper [6], the remainder with modulus n is constructed from remainders with moduli ($2n+1$) and ($2n+2$). In paper [7] the remainder with modulus n is constructed from remainders with moduli $2n$ and $2n+2$. Paper [8] proposes an efficient modulo p multiplication algorithm with moderate factors of ($p+1$) and ($p-1$).

In RSA cryptography, the modulus n is an odd composite number of the form $n = pq$ where p and q are two large odd primes. But the factorization of n is not known except for the decipherer. In El-Gamal cryptography the modulus is a prime number. In RSA when $n = pq$ and $p \equiv 1 \bmod 4$, $q \equiv 3 \bmod 4$ or vice-versa, then ($2n+2$) is a multiple of 8. In El-Gamal cryptography n is a prime p . If $p \equiv 3 \bmod 4$, then ($2n+2$) is a multiple of 8; when $p \equiv 5 \bmod 6$, ($2n+2$) is a multiple of 12. In RSA when $n = pq$ and $p \equiv q \equiv 3 \bmod 4$ or $p \equiv q \equiv 1 \bmod 4$ then ($2n+6$) is a multiple of 8 and can be

factored easily. In El-Gamal cryptography n is a prime p and when $p \equiv 1 \pmod{4}$ then $(2n+6)$ is a multiple of 8 and can be factored easily.

In this paper the New Modular Multiplication (NMM) algorithm is presented. In this algorithm the remainder with modulus n is constructed from remainders with moduli $(2n+2)$ and $(2n+6)$.

Section 2 presents the preliminary concepts including Chinese Remainder Theorem, RSA and El-Gamal Cryptosystems. Section 3 presents the new improvised modular multiplication theorem that gives the basis for the remainder computation together with numerical examples. Section 4 presents the NMM algorithm based on the modular computation derived in Section 3. The NMM technique serves as the basis for the exponentiation computation. Section 5 gives the performance of the new algorithm using estimation of the computational complexity and it is shown that this procedure is much faster than the direct calculation.

2. PRELIMINARIES

In this section, the Chinese Remainder Theorem, RSA Cryptosystem, El – Gamal cryptosystem and the exponentiation computation are presented.

2.1 Chinese Remainder Theorem [10]

Consider a system of congruences with different moduli, $x \equiv a_i \pmod{m_i}, i = 1, 2, 3, \dots, r$

Where $m_i, i = 1, 2, \dots, r$ are relatively prime. Then there exists a unique solution $x, 0 < x < M$ where

$M = m_1 m_2 \dots m_r$ and where x is computed as,

$$x = (a_1 M_1 N_1 + a_2 M_2 N_2 + \dots + a_r M_r N_r) \pmod{M}, \text{ Where } M_i = \frac{M}{m_i} \text{ and } N_i = M_i^{-1}$$

2.2 RSA Cryptosystem Algorithm [1]

The RSA cryptosystem uses computations in Z_n , where n is the product of two distinct odd primes p and q . Each user

- Chooses two large strong primes p and q . Let $n = pq$
- Computes $\Phi(n) = (p-1)(q-1)$
- Finds a random number e satisfying $1 < e < \Phi(n)$ and $\gcd(e, \Phi(n)) = 1$
- Computes a number d such that $d = e^{-1} \pmod{\Phi(n)}$.
- In RSA n and e are public keys and (d, p, q) are private keys. Plain text P is encrypted as
- $C = P^e \pmod{n}$. Cipher Text C is decrypted by $P = C^d \pmod{n}$

2.3 El – Gamal Cryptosystem [2]

- Fix a large finite field F_p^* where p is prime

- Each user A selects a generator $g \in \mathbb{F}_p^*$ and a where $0 < a < (p-1)$.
- User A makes public (g, g^a) . The integer a is the secret deciphering key.
- To send a message P to user A, User B chooses a secret integer k at random, $0 < k < (p-1)$
- User B computes $P(g^a)^k = Pg^{ak}$ and sends the pair (g^k, Pg^{ak}) to user A
- User A who knows the deciphering key a , recovers the plaint text P by computing $Pg^{ak}(g^k)^{-a} = Pg^{ak}g^{-ak} = P$.

2.4 Exponentiation Computation Algorithm [9]

The direct computation of $x^e \bmod n$ can be done using $(e-1)$ modular multiplications which is a tedious process for large values of e . The well-known Square – and – Multiply algorithm [9] reduces the number of modular multiplications required to atmost $2l$, where l is the number of bits in the binary representation of e .

3. The New Improvised Modular Multiplication Method

In this section the new modular multiplication algorithm is presented. In this algorithm remainder modulo n (where n is odd) is computed from remainders with moduli $(2n+2)$ and $(2n+6)$.

Theorem 3.1

In the set of integers, let n be a given odd positive integer. Let X denote any integer such that $0 \leq X \leq (n+2)^2$. Let $y = X \bmod n$, $y_1 = X \bmod (2n+2)$, $y_2 = X \bmod (2n+6)$. Then y can be expressed as

$$y \equiv \frac{(2y_1 - y_2)}{2} \bmod n \text{ if } y_1 \geq y_2$$

$$y \equiv \left\{ 3 + \frac{(2y_1 - y_2)}{2} \right\} \bmod n \text{ if } y_1 < y_2$$

Proof: Consider the equations,

$$y = X \bmod n \tag{1}$$

$$y_1 = X \bmod (2n+2) \tag{2}$$

$$y_2 = X \bmod (2n+6) \tag{3}$$

$$\text{From equation (2) there exists an integer } q_1 \text{ such that } X = (2n+2) q_1 + y_1 \tag{4}$$

From equation (3) there exists an integer q_2 such that

$$X = (2n+6) q_2 + y_2 \tag{5}$$

Multiplying equation (4) by $(2n+6)$ we obtain,

$$(2n+6) X = (2n+2) (2n+6) q_1 + (2n+6) y_1 \tag{6}$$

Multiplying equation (5) by $(2n+2)$ we obtain

$$(2n+2) X = (2n+2) (2n+6) q_2 + (2n+2) y_2 \tag{7}$$

From Equations (6) and (7) we obtain

$$X = (n+1)(n+3)(q_1 - q_2) + \left\{ \frac{(n+2)y_1}{2} - \frac{(n+1)y_2}{2} \right\} \quad (8)$$

Case 1: Suppose $(q_1 - q_2) < 0$

Since $y_1 \leq (2n+1)$ and $y_2 \geq 0$ it follows that,

$$\frac{(n+2)y_1}{2} - \frac{(n+1)y_2}{2} + (n+1)(n+3)(q_1 - q_2) < \frac{(n+2)}{2}(2n+1) - (n+1)(n+3) = -\frac{(n+2)}{2} < 0$$

Using equation (8), we conclude that $X < 0$, a contradiction, since $X \geq 0$. Hence $(q_1 - q_2)$ cannot be negative.

Case 2: Suppose $(q_1 - q_2) > 1$

Since $y_1 \geq 0$ and $y_2 \leq (2n+5)$ it follows that,

$$\frac{(n+2)y_1}{2} - \frac{(n+1)y_2}{2} + (n+1)(n+3)(q_1 - q_2) > -\frac{(n+1)}{2}(2n+5) + 2(n+1)(n+3) > (n+2)^2, \text{ a contradiction since } X \leq (n+2)^2.$$

Hence $(q_1 - q_2)$ cannot be greater than 1.

Since $(q_1 - q_2)$ is neither negative nor greater than 1, we conclude that $(q_1 - q_2) = 0$ or 1.

i.e., $q_1 = q_2$ or $q_1 = q_2 + 1$

When $q_1 = q_2$, on simplification we obtain $y_1 \geq y_2$: Using equation (8) we obtain

$$y \equiv \frac{(2y_1 - y_2)}{2} \pmod{n} \text{ if } y_1 \geq y_2.$$

(We observe that y_1 and y_2 are both odd if X is odd; y_1 and y_2 are both even if X is even. Hence $(2y_1 - y_2)$ is always even whether X is odd or even. Hence $\frac{(2y_1 - y_2)}{2}$ is an integer.)

When $q_1 = q_2 + 1$ on simplification we obtain $y_1 < y_2$: Using Equation (8) we obtain

$$y \equiv \left\{ 3 + \frac{(2y_1 - y_2)}{2} \right\} \pmod{n} \text{ if } y_1 < y_2$$

Thus $y = X \pmod{n}$ can be expressed as follows:

$$y \equiv \frac{(2y_1 - y_2)}{2} \pmod{n} \text{ if } y_1 \geq y_2 \quad (9)$$

$$y \equiv \left\{ 3 + \frac{(2y_1 - y_2)}{2} \right\} \pmod{n} \text{ if } y_1 < y_2 \quad (10)$$

Hence the result

Note:

In order to determine y it suffices to add or subtract n atmost thrice because the bounds for the right had side are $-(n+2)$ and $(3n+4)$.

The following examples illustrate theorem 3.1:

Example 3.1: Let $n = 23$ and $X = x^2$

When $x = 18$, $y_1 = X \bmod (2n+2) = 324 \bmod 48 = 36$,

$y_2 = X \bmod (2n+6) = 324 \bmod 52 = 12$.

Since $y_1 > y_2$, applying equation (9), we obtain $y = 2$

When $x = 21$, $y_1 = X \bmod (2n+2) = 441 \bmod 48 = 9$,

$y_2 = X \bmod (2n+6) = 441 \bmod 52 = 25$

In this case $y_1 < y_2$ Applying equation (10), we obtain $y = 4$

Example 3.2: Suppose in an RSA cryptosystem $n = 46927$, $X = x^2$,

Then $2n+2 = 93856$ and $2n+6 = 93860$

When $x = 16346$, $y_1 = x^2 \bmod (2n+2) = 267191716 \bmod 93856 = 77540$

$y_2 = x^2 \bmod (2n+6) = 267191716 \bmod 93860 = 66156$

Here $y_1 > y_2$, Applying equation (9), $y \equiv \frac{(2y_1 - y_2)}{2} \bmod n = 83232 \bmod 46927 = 36305$

It may apparently seem that the computation is made difficult because computations of $\bmod (2n+2)$ and $\bmod (2n+6)$ are used instead of a single computation of $\bmod n$. But we see later in Example 4.1 that computational complexity is greatly reduced when y_1 and y_2 are calculated using the Chinese Remainder Theorem.

4. MODULAR MULTIPLICATION ALGORITHM

In this section, we present the **NMM** algorithm for the new modular multiplication.

The exponentiation computation is composed of the modular multiplication $x^2 \bmod n$ and $xu \bmod n$. It was shown in Section 3 that $y = X \bmod n$ can be obtained from $y_1 = X \bmod (2n+2)$ and $y_2 = X \bmod (2n+6)$. Algorithm **NMM** calculates y_1 and y_2 using the Chinese remainder theorem. This enhances the computational speed.

As a first step $(2n+2)$ and $(2n+6)$ are decomposed into products of mutually prime factors. Note that this decomposition need not necessarily be the prime factorization.

$$2n+2 = \prod_{i=1}^r p_i$$

$$2n+6 = \prod_{i=1}^s q_i$$

Next, the following algorithm receives x, u such that $u, 0 \leq x \leq (n+2)^2, 0 \leq u \leq (n+2)^2$ and outputs $y = xu \bmod (p_1 p_2 \dots p_r)$

Algorithm NMM(x, p, y)

Input: $x, u, 0 \leq x \leq (n+2)^2, 0 \leq u \leq (n+2)^2, p = (p_1 p_2 \dots p_r)$

Output: $y = xu \bmod (p_1 p_2 \dots p_r)$

Step 1: Calculate $x_i = x \bmod p_i, u_i = u \bmod p_i, i = 1, 2, \dots, r$

Step 2: Calculate $a_i = x_i u_i, i = 1, 2, \dots, r$

Step 3: Calculate $a_i = a_i \bmod p_i, i = 1, 2, \dots, r$

Step 4: Calculate y by Chinese Remainder Theorem algorithm

Using Algorithm NMM, $y_1 = x^2 \bmod (2n+2)$ and $y_2 = x^2 \bmod (2n+6)$ are obtained by NMM (x, p, y_1) and NMM (x, q, y_2) .

Example 4.1: Consider the same RSA cryptography as in Example 3.2

Let $n = 46927$, $(2n+2)$ and $(2n+6)$ are decomposed into relatively prime factors

$$2n+2 = 93856 = 7 \times 32 \times 419, 2n+6 = 93860 = 20 \times 13 \times 361$$

Let, $p_1 = 7, p_2 = 32, p_3 = 419; q_1 = 20, q_2 = 13, q_3 = 361$.

Let $x = 16346$ and suppose $y = x^2 \bmod n$ is to be calculated. The computational procedure for NMM $(16346, 7 \times 32 \times 419, y_1)$ is shown below:

Step 1: $x_1 = 16346 \bmod 7 = 1, x_2 = 16346 \bmod 32 = 26, x_3 = 16346 \bmod 419 = 5$

Step 2: $a_1 = 1^2 = 1, a_2 = 26^2 = 676, a_3 = 5^2 = 25$

Step 3: $a_1 = 1 \bmod 7 = 1, a_2 = 676 \bmod 32 = 4, a_3 = 25 \bmod 419 = 25$

Step 4: Solving the following system of congruence equation,

$$y_1 \equiv 1 \bmod 7, y_1 \equiv 4 \bmod 32, y_1 \equiv 25 \bmod 419,$$

Using Chinese Remainder Theorem, we get $y_1 = 77540$

Similarly $y_2 = 66156$ is obtained from NMM $(16346, 13 \times 20 \times 361, y_2)$ and by solving the system of congruence equations, $y_2 \equiv 12 \bmod 13, y_2 \equiv 16 \bmod 20, y_2 \equiv 93 \bmod 361$,

$$\text{From } y_1 = 77540 \text{ and } y_2 = 66156, \text{ we obtain } y = \frac{(3y_1 - y_2)}{2} \bmod n = 83232 \bmod 46927 = 36305,$$

Note that the value is the same as in example 3.2.

5. COMPUTATIONAL COMPLEXITY OF NMM ALGORITHM

In this section, the computational complexity of NMM algorithm is compared with that of the ordinary direct method. The complexity computation takes into account only the operations of multiplications and divisions. The standard bit computational complexity is considered. Then computational complexities for multiplication and division are given respectively as follows:

$M(k, l)$ = computational complexity for $k \times l$ bit number = $l(k+l)$ and

$D(k, l)$ = computational complexity for k / l bit number = $l(k-l)$

The computational complexity in the preliminary computation is not included. In the following discussion n is assumed to consist of k bits and p_1, p_2, \dots, p_r and q_1, \dots, q_s of l bits at the maximum.

Complexity of the Ordinary Direct Method

In the ordinary calculation of $x^2 \bmod n$, the multiplication of two k -bit numbers and the division of a $2k$ -bit number by a k -bit number are required. The computational complexity for this is $T_0 = m(k, k) + D(2k, k) = 3k^2$

Complexity of the New Method

To calculate the right hand side of the system of congruence equations the division of the k -bit number by l -bit number is required in stage 1, the multiplication of two l -bit numbers is required in step 2 and the division of $2l$ -bit number by l -bit number is required in step 3. Adding, the computational complexity for each factor i is $D(k, l) + M(l, l) + D(2l, l) = (k-l) + 2l^2 + l^2 = l(k+2l)$, which is needed for each i .

In solving the system of congruence equations, as in example of 4.1 Step 4, the multiplication of the l -bit number and the k -bit number as well as the division of the $(k+l)$ bit number by k -bit number are required. Then the total is $M(k, l) + D(k+l, l) = l(2k+l)$.

Thus the computational complexity of the proposed method is $T = 3l(k+l)$.

As a special case, when $l = k/2$ as in the previous numerical example, $T = 9/4k^2$ which is $3/4$ of the direct method. When $l = k/K$, $T = ((K+1)/K^2) T_0$. When K is further increased the computational complexity approaches $1/K$.

The smaller the prime factors of $2n+2$ and $2n+6$ greater the efficiency of the NMM algorithm

6. CONCLUSIONS

In this paper, it is shown that remainder modulo n can be determined from remainders with moduli $(2n+2)$ and $(2n+6)$ by decomposing $(2n+2)$ and $(2n+6)$ into relatively prime factors for the case where $n = pq$, $p \equiv q \equiv 3 \pmod{4}$ and $p \equiv q \equiv 1 \pmod{4}$ and when n is a prime p such that $p \equiv 1 \pmod{4}$. Also it is shown that the remainders moduli $(2n+2)$ and $(2n+6)$ can be calculated using the Chinese Remainder Theorem and modular multiplication can be realized with lesser computational complexity. The efficiency of the NMM algorithm is greater when the prime factors are smaller. The method proposed in this paper is useful in exponentiation based cryptography and also in improving the speed of signal processing that requires similar computation.

REFERENCES

1. Rivest R, Shamir A, Adleman L, "A method for obtaining digital signature and public key cryptosystems", communications of the ACM, Vol. 21, pp 120-126, 1978.
2. El Gama1 T, "A Public key cryptosystem and signature scheme based on discrete logarithms," IEEE Transactions on Information Theory, Vol. 31, pp 469-472, 1985.
3. Rabin M.O, "Digital Signatures and public key functions as intractable as Factorization," MIT/LCS/TR-212, MIT Laboratory for computer science, 1979.
4. Diffie W, Hellman M, "New directions in cryptography", IEEE Transactions on Information Theory, Vol. 22, pp 644-654, 1976.

5. Hyashi. A, A new fast modular multiplication method and its application to modular exponentiation based cryptography," Electronics and communications in Japan, Vol. 83, pp 88-93, 2000.
6. G.A.V. Rama Chandra Rao, P.V. Lakshmi and N. Ravi shankar, "A New Modular Multiplication Method in public key cryptosystem," International Journal of Network Security, Vol. 15, No.1, pp 23-27, Jan. 2013.
7. G. A. V. Rama Chandra Rao, P. V. Lakshmi and N. Ravi Shankar, "A Novel Modular Multiplication Algorithm and its Application to RSA Decryption," IJSCI International Journal of Computer Science issues, Vol. 9, Issue 6. No. 3, November 2012 ISSN (online): 1694-0814.
8. Hwang R.J, Su F.F and Shiau. S. H," An Efficient modulo p multiplication Algorithm with moderate factors of $(p+1)$ and $(p-1)$," Communications of Mathematics Science, Vol. 5, No. 2, pp383-389, 2007.
9. Knuth DE.," The art of computer Programming" Vol. 2, 2nd Addisom Wesley," 1980
10. KoblitzN A Course in Number Theory and Cryptography; Springer; 1987, Chapter 1.